Software Testing Documentation

Comprehensive Database Tests

Each of the following tests were conducted incrementally, starting with an empty database file. The success of one test determined the starting criteria for the next. All database tests were written in a single test file, data_test.py, and utilizes data.py, file.py, and url.py.

Author of data_test.py: Tim Yingling

Test #	Description	Preconditions / Input Values	Expected Output	Actual Output	P/F Criteria	Comments
1	Test to see if the newly created database file has empty tables	Database file and tables for files and urls are created, use print_all_files and print_all_urls methods on empty tables.	Nothing is printed except a newline character after the program declares that it is printing the contents of each table	Nothing is printed except a newline character after the program declares that it is printing the contents of each table	Actual = Expected	Pass
2	Test to see if store_file and store_url methods insert the passed objects into an empty database correctly. Store_file is passed a File object and stores	f1 = File(“Tim”, “Mark”, “Clement”, “Jenna”) u1 = URL(domain_name = “Google”) store_file(f1) and store_url(u1) are	Two tuples displayed showing the contents of f1 and u1 respectively: (‘Tim’, ‘Mark’, ‘Clemont’, ‘Jenna’, 0, 0, 0, 0, ‘’, ‘’) (‘’, ‘’, ‘’, ‘’, 0, ‘’, ‘’, ‘’, ‘Google’, 0)	Two tuples displayed showing the contents of f1 and u1 respectively: (‘Tim’, ‘Mark’, ‘Clemont’, ‘Jenna’, 0, 0, 0, 0, ‘’, ‘’) (‘’, ‘’, ‘’, ‘’, 0, ‘’, ‘’, ‘’, ‘Google’, 0)	Actual = Expected	Pass

	all of the member variables into the files table. Store_url is passed a URL object and stores all of the member variables into the urls table.	called with an empty database				
3	Test to see if search_file and search_url methods return None if the item is not found in the database. Search_file is passed a string and is checked against any of the hash types stored to find a match. Search_url is passed a string that is checked against domain names stored to find a match.	Database has f1 and u1 in their respective tables. string1 = “Biruk string2 = “Amazon” search_file(string1) and search_url(string2) are called on the database	Both functions return None and “File not found” and “URL not found” are displayed in the terminal	Both functions return None and “File not found” and “URL not found” are displayed in the terminal	Actual = Expected	Pass
4	Test to see if store_file and	f2 = File(“Biruk”, “Scott”, “Nick”,	Four tuples displayed showing the contents of	Four tuples displayed showing the contents of f1,	Actual = Expected	Pass

	store_url methods insert the passed objects into an existing database correctly	“Kyle”) u2 = URL(domain_name = “Amazon”) store_file(f2) and store_url(u2) are called on the database.	f1, f2, u1, and u2 respectively: (‘Tim’, ‘Mark’, ‘Clemont’, ‘Jenna’, 0, 0, 0, 0, ‘’, ‘’) (‘Biruk’, ‘Scott’, ‘Nick’, ‘Kyle’, 0, 0, 0, 0, ‘’, ‘’) (‘’, ‘’, ‘’, ‘’, 0, ‘’, ‘’, ‘’, ‘Amazon’, 0)	f2, u1, and u2 respectively: (‘Tim’, ‘Mark’, ‘Clemont’, ‘Jenna’, 0, 0, 0, 0, ‘’, ‘’) (‘Biruk’, ‘Scott’, ‘Nick’, ‘Kyle’, 0, 0, 0, 0, ‘’, ‘’) (‘’, ‘’, ‘’, ‘’, 0, ‘’, ‘’, ‘’, ‘Amazon’, 0)		
5	Test to see if File and URL objects are returned correctly after calling search_file and search_url on items that exist in the tables	Database has f1, f2, u1, and u2 in their respective tables. string1 = “Biruk string2 = “Amazon” search_file(string1) and search_url(string2) are called on the database	Search_file(string1) returns File object similar to f2 and search_url(string2) returns URL object similar to u2. “File found, md5 hash = Biruk” and “URL found, domain name = Amazon” displayed to terminal	Search_file(string1) returns File object similar to f2 and search_url(string2) returns URL object similar to u2. “File found, md5 hash = Biruk” and “URL found, domain name = Amazon” displayed to terminal	Actual = Expected	Pass
6	Test to see if search_file will find an existing file using a different hash	Database has f1, f2, u1, and u2 in their respective tables. String = “Scott”	Search_file(String) returns a File object similar to f2 “File found, md5 hash =	Search_file(String) returns a File object similar to f2 “File found, md5 hash = Biruk” displayed to	Actual = Expected	Pass

	type than md5 which is the first hash type stored in the table	search_file(String) called on the database	Biruk” displayed to terminal	terminal		
7	Test to see if search_URL can pull from the database and get the appropriate engines and results	Database has f1, f2, u1, and u2 in their respective tables. string = URL	search_URL(string) returns File object with correct engines and results	search_URL(string) returns File object with incorrect engines and results	Actual = Expected	Fail For the URL class, the parameters “engines” and “results” needed to be switched because the database was retrieving the correct info but storing them in the wrong member variables
8	Test to see if database adds URLs that have the same domain but are different sites	Database has f1, f2, u1, and u2 in their respective tables. database has site with same domain as passed-in URL string = URL	search_URL(string) returns None when the string has the same domain as another in the database but is a different site; store_URL is called	search_URL(string) returns a File object because it searches the database based on the domain which match between the passed-in string and the string in the database	Actual = Expected	Fail Need to search database based on what the clean user input is and not domain name because different sites may have the same domain name, but they are technically different sites so they must be scanned and added separately

Comprehensive Frontend Blackbox Tests

Each of the following tests were conducted incrementally and independently. Each feature on the front-end was tested thoroughly with their respective output described below.

Author of test: Mark Biegel

Test #	Description	Input Values	Expected Output	Actual Output	P/F Criteria	Comments
1	Front-end accounts for all types of hash and URL malicious status including clean, suspicious, and malicious	A URL and hash of the following malicious statuses: clean, suspicious, and malicious	A checkmark for a clean hash A caution sign for suspicious hash An x-mark for a malicious hash	A checkmark for a clean hash An x-mark for a malicious hash	Actual = Expected	Fail VirusTotal returned a “suspicious” flag, but the frontend doesn’t account for it.
2	Front-end accounts for invalid hash or URL and a query limit of 500 new queries per day	Non-urls, non-hashes, or more than 500 new queries	Rendered “invalid_request.html”	Rendered “invalid_request.html”	Actual = Expected	Pass
3	Front-end redirects to “help.html” when “Help” button is clicked	“Help” button on “homescreen.html” is clicked	Rendered “help.html”	Rendered “help.html”	Actual = Expected	Pass

4	Front-end redirects to “about.html” when “About” button is clicked	“About” button on “homescreen.html” is clicked	Rendered “about.html”	Rendered “about.html”	Actual = Expected	Pass
5	Front-end redirects to “homepage.html” whenever the Threat Detector logo is clicked except for on the homepage	Logo icon on any .html page is clicked	Rendered “homepage.html”	Rendered “homepage.html”	Actual = Expected	Pass
6	Front-end redirects to the second page html templates when “Search” button is clicked on the homepage or on the second page templates.	Search button is clicked on homepage or second page templates	Rendered “second_page_hash_malware_negative”, “second_page_hash_malware_positive”, “second_page_URL_malware_negative”, “second_page_URL_malware_positive”	Rendered “second_page_hash_malware_negative”, “second_page_hash_malware_positive”, “second_page_URL_malware_negative”, “second_page_URL_malware_positive”	Actual = Expected	Pass
7	Front-end displays the Detection tab when click on either of the second page templates	Detection button is clicked on the second page templates	Detection tab is displayed with the list of vendors and detection status	Detection tab is displayed with the list of vendors and detection status	Actual = Expected	Pass

8	Front-end displays the Metadata tab for a URL or hash when click on either of the second page templates	Metadata button is clicked on the second page templates	Metadata tab is displayed with the list of statistics and values for either a URL or hash	Metadata tab is displayed with the list of statistics and values for either a URL or hash	Actual = Expected	Pass
9	Front-end displays the Malicious Status tab when click on either of the second page templates	Malicious Status button is clicked on the second page templates	Malicious Status tab is displayed with a description of the malicious status for either a URL or hash	Malicious Status tab is displayed with a description of the malicious status for either a URL or hash	Actual = Expected	Pass